схемы как совокупности логических элементов и связей между ними, необходимо показать графическую интерпретацию и практическое применение: фиктивной переменной; суперпозиции булевых функций; лемм о несамодвойственной, немонотонной и нелинейной функциях; реализации констант 0 и 1, отрицания, конъюнкции на логических элементах функции, представляющей функционально полный класс. А в среде Quartus II, при моделировании конкретных цифровых устройств, закрепить полученные знания.

Данный подход выводит преподавание предмета на качественно иной уровень, делает более доступным для понимания и наглядным для понимания теоретический материал булевых функций.

Библиографический список

1. Исмагилова Е.И. Булевы функции и построение логических схем: учебное пособие / Е.И. Исмагилова – М.: МИРЭА, 2015. 160 с.

2. Исмагилова Е.И. Краткий обзор истории развития геометрических методов логики // Актуальные проблемы истории естественно-математических и технических наук и образования: Материалы Всероссийской научно-практической конференции. Елабуга, 2014. С.82-90.

УДК 004.43
ББК 32.973.2

Кобелев И.А.
Елабужский институт КФУ, г. Елабуга
kobelevia@yandex.ru

ЧТО ТАКОЕ АЛГОРИТМ?

Аннотация. Понятие алгоритма является одним из фундаментальных понятий как информатики, так и математики. Теория алгоритмов изучает общие подходы к решению проблем обработки дискретной информации, что не маловажно для будущих специалистов в области прикладной информатики и математики, специалистов в области программирования. Поэтому теория алгоритмов является теоретическим фундаментом изучения информатики и вычислительной техники. В работе рассматривается подход к понятию алгоритма и начальная классификация языков программирования.

Ключевые слова: алгоритм, программирование, языки программирования.

В работе рассматривается подход к понятию алгоритма и начальная классификация языков программирования (ЯП).

Наука, заявляющая о собственной всеобщности, а именно – философия – дает следующее "определение определения":

Чтобы определить нечто надо:

- *указать класс объектов, к которому принадлежит определяемое;*
- *указать отличие определяемого элемента от других в классе.*

Так СТОЛ определяем как вид мебели, предназначенной для работы.

Нетрудно сообразить, что под это определение подходят и верстак, и ученическая парта, и профессорская кафедра, и кухонный стол, ложкой тоже надо работать.

Понятие СТОЛ оказалось определенным через понятие МЕБЕЛЬ, которое, в свою очередь, хотелось бы тоже определить, но это будет через новое понятие. Цепочка бесконечна!

Науки, называющие себя ТОЧНЫМИ, рвут эту цепочку, признавая некоторую группу объектов неопределяемыми:

– в теории множеств – множество;

– в геометрии – точка, прямая, плоскость;

– в физике – материальная точка;

...

Учитель геометрии в школе ставит пятно мела на доске и говорит:

"Берем точку А ...".

Насколько прав учитель? Является ли пятно мела на доске точкой? Учитель абсолютно прав.

Неопределяемые в некоторой теории понятия связаны АКСИОМАМИ.

В геометрии имеется аксиома "через любые две точки проходит единственная прямая". С помощью деревянной линейки для любых двух пятен мела на доске стоим единственную соответствующую линию. Можем так же убедиться в справедливости для "пятен мела" других аксиом геометрии, связанных с точкой. Раздел геометрии, рассматривающий пятна мела и чернил (грифеля) как образцы точек имеет собственное название – начертательная геометрия.

Точно так же, точкой можно считать пару чисел (декартовы координаты), прямая отождествляется с соответствующим уравнением. Единственности можно добиться, потребовав равенству единице коэффициента при y, или рассматривая уравнение "в отрезках". Раздел геометрии – планиметрия.

В стереометрии, точкой считают тройку чисел. Проективная геометрия получается при изменении равномерности масштаба координат. Когда координатные оси нелинейны имеем дифференциальную геометрию.

Планиметрия, стереометрия, начерталка – считаются моделями геометрии.

Какой бы ни была геометрия, "точки" и "прямые" в ней связаны одним и тем же набором аксиом и полученные в одной из геометрий результаты, не связанные нечем, кроме аксиом, переносятся в другую геометрию. По сути, геометрия оказалась одна.

Еще одно замечание: Основная, ныне действующая в геометрии аксиоматика, базируется на предложениях Евклида, приняла окончательный вид под пером Гильберта, ее эквивалентами являются адаптированные для школьников аксиоматики Киселева и Погорелова. В этих аксиоматиках понятие ВЕКТОР определяется через понятие ОТРЕЗОК, определенный из понятия ПРЯМАЯ. Существует, предложенная Вейлем аксиоматика геометрии, где неопределяемыми предлагается выбрать всего два понятия: ТОЧКА и ВЕКТОР. Тогда прямой будет множество концов векторов, параллельных направляющему, отложенных от некоторой точки. Параллельные вектора получаются как результат произведения действительного числа на заданный вектор. Для определения точек плоскости, от данной точки откладываем вектор – линейную комбинацию двух векторов, пространства – трех.

Можно строго доказать соответствие получившихся геометрий: что у Вейля аксиомы, у Гильберта теоремы и наоборот. В отличие от "гильбертовской", аксиоматику Вейля удобней распространить на многомерные пространства. Выполнение задач компьютерной графики так же удобней решать, представляя прямые и плоскости как точку и определяющие векторы.

Какое отношение вышесказанное имеет к алгоритмам?

Как догадался читатель, автор настаивает на том, чтобы понятие алгоритма отнести к неопределяемым. В работах Гейна имеется заявление, что предлагаемое в школьных учебниках "определение алгоритма" ничем не хуже, чем целый ряд базовых определений в других науках. Но беда в том, что в отличие от этих "других" наук, информатика (теория алгоритмов, как ее часть) претендуют (и не без основания!) на звание ТОЧНЫХ наук. И поэтому, желательно, соответствующее отношение.

В точных науках, как показано выше, неопределяемые понятия косвенно определяются системой аксиом, типа:

"Точка – то, что удовлетворяет соответствующим аксиомам".

И для алгоритма уже есть такие аксиомы. Это – основные свойства алгоритмов.

Если задуматься, определение алгоритма типа *"упорядоченный набор действий, за конечное число шагов приводящий к результату"* есть перечисление основных свойств алгоритма.

Собрать все свойства алгоритма в короткое предложение не получается, и, рассматривая популярную литературу по информатике, можно найти десятки разных "определений" понятия АЛГОРИТМ. Но, если считать основные свойства алгоритма – аксиомами, нужно так же подойти к ним, то есть исследовать на полноту и непротиворечивость.

В связи с этим, мне кажется уместным, главной аксиомой считать

ДВОЙСТВЕННОСТЬ

- алгоритм можно и записать, и исполнить.

Есть в школьном списке "наличие исполнителя", но мне кажется (см. пример ниже) предлагаемая формулировка лучше.

Почему-то в учебниках (и не только школьных) на этот аспект обращается мало внимания. Ведь в ЯП basic есть отдельные команды RUN и LIST, для исполнения и просмотра записи алгоритма.

Следующей аксиомой можем считать

ДИСКРЕТНОСТЬ

- алгоритм может быть разбит на отдельные шаги, каждый шаг можем считать более простым алгоритмом, группу отдельных шагов можем считать одним шагом более сложного алгоритма. (Аналогичная аксиома определяет понятие МНОЖЕСТВО в теории множеств.)

ДЕТЕРМИНИРОВАННОСТЬ (ОПРЕДЕЛЕННОСТЬ)

- всегда известно (определено), какой шаг следует выполнять как очередной. При таком понимании, присутствие свойств алгоритма "наличие начала", "наличие конца" в списке основных свойств алгоритмов не обязательно.

МАССОВОСТЬ

- алгоритм предназначен для решения целого класса задач, различающихся, например, к исходными данными.

РЕЗУЛЬТАТИВНОСТЬ

- алгоритм выдает результат за конечное число шагов. Считаю целесообразным свойство "конечность" объединить с "результативность". Эта аксиома порождает класс алгоритмически неразрешимых задач, для которых примененный алгоритм не остановится.

ОДНОЗНАЧНОСТЬ

- при одинаковых исходных данных, алгоритм выдает один и тот же результат. Следует различать цели алгоритмов, использующих и не использующих случайные числа, то есть требуется конкретизировать, что считается результатом.

Обещанный пример:

Является ли алгоритмом биение сердца? Если да, то почему этим занимаются медики? Если нет, то, какое свойство алгоритма отсутствует?

ОДНОЗНАЧНОСТЬ и РЕЗУЛЬТАТИВНОСТЬ присутствуют:
- биение сердца поддерживает жизнь организма, при изменении исходных данных (попадание тромба) имеем другой результат.

МАССОВОСТЬ присутствует:

- имеем толпы людей, у каждого, физиологически, имеется сердце. Кроме того, сердце имеется у зверей, птиц, рыб...

ДЕТЕРМИНИРОВАННОСТЬ и ДИСКРЕТНОСТЬ присутствуют:
- специалисты различают отдельные фазы работы сердца, расширение и сокращение, которые следуют друг за другом.

ДВОЙСТВЕННОСТЬ вот тут собака и зарыта:

- мы можем записать биение сердца, например на кардиограмму, но заставить сердце биться согласно имеющейся кардиограмме – нет.

Одно из свойств алгоритма (кстати, отсутствующее в учебниках) нарушено, что и является причиной того, что именно медики (терапевты, кардиологи), в основном занимаются биениями сердца.

Приведенный (аксиоматический) подход к понятию алгоритма считается интуитивным. Набор основных свойств (аксиом) как бы предложен "свыше". Но в информатике есть свои "Вейли", которые предложили конструктивный подход.

Суть конструктивного подхода – вывод понятия "алгоритм" из понятия "функция".

Понятие ФУНКЦИЯ определяется в рамках теории множеств примерно так:

МНОЖЕСТВО – неопределяемое понятие.

ПРОИЗВЕДЕНИЕ МНОЖЕСТВ (декартово) – операция порождения новых множеств из существующих.

ОТНОШЕНИЕ – любое подмножество декартового произведения множеств.

Некоторые отношения обладают СВОЙСТВАМИ (рефлексивность, симметричность,...). Есть свойства, выделяющее функциональные отношения, обозначения которых и есть ФУНКЦИИ.

Задать функцию можно одним из трех способов:

По определению – ФУНКЦИЯ – частный случай отношения и может быть задана *таблицей*. В геометрии предпочитают задавать функцию *графиком*. Остальные математики задают функции *формулами.*

Отказываясь от теории множеств, находясь в рамках математического анализа, к понятию функции вынуждены относиться, как к неопределяемому. Исследование вопроса "что такое функция" привело к так называемым вычислимым функциям.

Прежде всего, для максимального упрощения, к вычислимым функциям относят только те, у которых и область определения, и область значения – только натуральные числа. Исключение делается для числа 0. Далее это множество обозначаем N0 (иногда его обозначают Z>=0 – целые неотрицательные).

Функция называется вычислимой, если существует АЛГОРИТМ, который из исходных данных выдает значение.

Считая понятие функции известным (из теории множеств), перевернем вышестоящее определение:

Алгоритм – это то, как вычислимая функция из аргументов получает значение.

Является ли это определение определением алгоритма? Ведь алгоритмы работают с чем угодно, а вычислимые функции – с элементами из N0?

Этот вопрос решает теория конструктивности, которая из пары натуральных получает целые числа, из пары целых – рациональные. Теория сечений (пределы) приводит к действительным числам, пара которых – комплексное, четверка – кватернион, восьмерка – октет.

Другая наука – теория нумераций, произвольным объектам, взаимно-однозначно, ставит в соответствие номера. То есть по полученному вычислимой функцией результату, фактически, можем иметь что угодно: и график функции, и табуретку (получается соединением деталей №1, №2, ... по технологическому плану). Осталось только добавить о наличии возможности оцифровки изображений и звука.

При исследовании механизма, как все-таки вычислимые функции дают результат, была сформулирована Машина Тьюринга (МТ). Если у нее исходные данные и итог – из N0, имеем реализацию (модель) вычислимых (по Тьюрингу) функций. Расширяя внешний алфавит МТ, имеем возможность работы с произвольной знаковой информацией.

Сужая внешний алфавит МТ до двух знаков ("пробел" и "черточка" == 0 и 1) имеем машину Поста. Приближая МТ к реальности (ограничиваем ленту хотя бы в одну сторону) имеем RAM-машину. Наконец, приступая к практической реализации вычислений, имеем компьютер.

Алгоритмы для компьютера, согласно свойству двойственности, должны быть заранее записаны, такую запись называют программой. Способы таких записей называют ЯП.

ЯП, базирующиеся на принципах машины Тьюринга, обобщенных фон-Нейманом, называют *алгоритмическими* (=процедурными).

Другая попытка реализации вычислимых функций – теория рекурсивных функций, в основе которой лежат исследования Черча, называемые иногда лямбда-исчислением. ЯП, в основе которых лежит эта теория называют *функциональными* (ФЯП).

Работа программы как функции, связана с получением исходных данных и выдачей результата. Исходные данные могут быть получены другой функцией, в математике это называется суперпозицией. Поскольку функция выдает только один результат, понятие переменной в ФЯП сужается: переменная не может менять полученный результат, служит только как перенос информации от одной функции в другую. Особенность ФЯП – представление исходных данных в виде списка аргументов функции, требует отнесения типа данных "список" к базовым (базовый тип данных – ситуация, когда заранее решены вопросы о размере области памяти для хранения некоторой информации, о способе ее кодирования, о списке возможных с ней действий и их реализации.). Это, в свою очередь, позволяет считать всю программу списком выполняемых функций.

Как понятие точки не зависит от применяемой модели (см. выше планиметрию, стереометрию и другие) так и понятие алгоритма не должно зависеть от подхода к понятию алгоритма. То есть, задачи, решаемые алгоритмическими ЯП должны решаться и ФЯП и наоборот. Это заявление зашифровано в известных тезисах Черча и Тьюринга.

Для организации ФЯП, потребовался отказ от понятия переменной как ячейки ленты МТ, это произошло сравнительно недавно. В настоящее время алгоритмических ЯП насчитывается сотни, если не тысячи, ЯП, относящихся к функциональным не так много. Первым считается Lisp, приложение «информатика» к «1 сентября» рекомендует Haskel, ...

Основной проблемой при использовании ФЯП является то, что он вынужден работать под управлением операционной системы, реализованной по правилам алгоритмических ЯП. Попыткой исправить эту ситуацию является создание "лисп-машин".

Таким образом, все ЯП сводятся в две большие группы – алгоритмические и функциональные. Существует ЯП "prolog", который, вроде-бы не вписывается в предложенную схему. Но, если рассматривать предикаты, из которых состоит пролог-программа, как функции, выдающие логический результат, получаем пролог – частный случай ФЯП, работающий с узким классом только логических функций.

УДК 378:514
ББК 74.58+22.151

Костин А.В., Костина Н.Н.,
Елабужский институт КФУ, г. Елабуга,
kostin_andrei@mail.ru, natnikost@mail.ru
Миннегулова Е.О., Сиразов Ф.С.
НГПУ, г. Набережные Челны,
e.o.minnegulova@gmail.com,fsirazov@yandex.ru

ИЗУЧЕНИЕ НЕЕВКЛИДОВЫХ ГЕОМЕТРИЙ С ИСПОЛЬЗОВАНИЕМ КОМПЬЮТЕРНЫХ ПАКЕТОВ

Аннотация. В работе обсуждаются проблемы подготовки будущих учителей математики. В качестве средства, помогающего компенсировать недостаток времени на освоение учебных дисциплин, предлагается использование свободно распространяемых систем компьютерной алгебры.

Ключевые слова: неевклидовы геометрии, системы компьютерной алгебры, компетентностный подход, имитационное моделирование.

Системы компьютерной алгебры помогают существенно оптимизировать учебный процесс. Необходимость такой оптимизации обусловлена в первую очередь сокращением аудиторных занятий по дисциплинам, которое произошло во многих высших учебных заведениях, и одновременным общим увеличением аудиторной нагрузки преподавателей. Конечно, основной целью обучения математике является развитие мышления вообще и в предметной области в частности. Декларируемый в последние годы компетентностный подход как будто бы полностью индифферентен к предметным знаниям, но невозможно всё мочь и уметь, ничего не зная. К тому же при всевозможных централизованных проверках, аккредитациях тестирование студентов проводится с оцениванием главным образом «знаниевой компоненты»